



# Computer modelling and natural phenomena

Giuseppe Longo

## ► To cite this version:

Giuseppe Longo. Computer modelling and natural phenomena. 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering, 2003, Helsinki, Finland. 10.1145/940071.940072 . hal-03320087

**HAL Id: hal-03320087**

**<https://hal-ens.archives-ouvertes.fr/hal-03320087>**

Submitted on 16 Aug 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Computer modelling and natural phenomena

**Author:** [Giuseppe Longo](#)

[Ecole Normale Supérieure, Paris et CREA, Ecole Polytechnique](#)

## **Published in:**

[ESEC/FSE-11](#) Proceedings of the 9th European software engineering conference held jointly with  
11th ACM SIGSOFT international symposium on Foundations of software engineering

Pages 1 - 5

[ACM](#) New York, NY, USA ©2003

ISBN:1-58113-743-5 doi>[10.1145/940071.940072](#)

# Computer Modelling and Natural Phenomena\*

Giuseppe Longo  
CNRS et Dépt. d'Informatique  
École Normale Supérieure, Paris  
et CREA, École Polytechnique  
+33 1 44 32 33 28  
longo@di.ens.fr

## ABSTRACT

This provocative synthetic introduction to several research themes aims at stimulating a reflection on our mature science, Informatics, beyond the myths that originated it and that, today, may affect its progress. The awareness of the expressiveness and of the internal limitations of digital computing is a necessary step, within the Computer Science community, to improve the relations to other sciences, where computers are increasingly used as tools. We will briefly hint to simulations problems in Physics and discuss the still prevailing projection of our fantastic machine onto Biological and Cognitive phenomena. The conference lecture will mostly focus on §.3.

## 1. CERTAINTY, EXACTNESS AND ABSOLUTES

Computer Science is changing the world by the machines it gives us, by the tools for thought these represent and by the image of the world it proposes. No scientific experience can be done today without computing as a tool, as a model and as a component of the very physical measure.

Our computing machinery derived from a robust *philosophical* project, which originated in the work by Boole and Frege, and was continued, on different grounds, by Hilbert. The idea was to found mathematics (indeed, knowledge) on Arithmetic, the locus for logic, certainty and effectiveness. The absolute notion of number (Frege: “an absolute concept”) and the finite structure of formal - potentially mechanizable - deductions (Hilbert) could found Mathematics and knowledge, independently of the ambiguities of meaning and reference to space. In particular, certainty could be reached away from the ongoing revolution in the geometrisation of Physics in non-euclidean continua, those of Riemann’s Geometry, whose intuitive meaning lead to a “delirium” - said Frege, 1884. And the new foundational analysis were entirely grounded on a philosophy of Arithmetic.

The 1930’s gave us the arithmetic machine Hilbert had dreamed of, the actual machine for his intended “potential mechanization” of formal proofs. The encoding of formal

languages into numbers and, by this, of metatheories into arithmetical theories, was Gödel’s fantastic invention, which Turing extended to a coding of the world into the tape of his machine. A machine, which started our discipline by another fundamental idea: Turing’s Logical Computing Machine of ‘35 is based on the distinction between software and hardware (both as purely mathematical concepts, at the time). Formal deductions were thus reduced to (encoded in) the least steps of “a man in the least act of thought”: move right, left, erase/write 0 or 1, over a (paper) hardware.

The subsequent physical realisation of this abstract machine fully inherited the software/hardware split, the exactness of arithmetic, the absolute access to the data and the certainty of the stepwise procedures: they are electronic *arithmetic* machines. The software/ hardware distinction allowed Programming Design and Theory to be conceived independently of hardware. Portability is the motto of all our applications: in particular, when the hardware of a machine is dying, one may transfer all its software into another.

Formal Logic and Arithmetic gave us a fantastic, but absolutely artificial tool: they engendered the highest invention of human mind, so original that nothing in the world resembles to it. But the alphabet as well, which radically departed from early ideographic representation of language, or the marvellous clocks of the XVIII century, with their clogs and pulleys, do not exist in nature. Yet, Aristotle claimed that thought is the impressing in the body of alphabet letters, like on a wax-tablet ([8]), and XVIII century philosophers saw in pulleys and clogs the essence of living organs, including brain. We always projected our latest invention onto nature. And still today, some claim that the digital computer is the “final machine”, the one that realises Frege’s logic and, thus, the inherent mechanics of Nature, including life and thought. That is, this time, we are done: this tool truly is an image of the world. It thinks for us, its encoding techniques explain genome; programming can elucidate evolution; the brain may be understood as, or actually is, a programmable computer.

Yet, progress in scientific knowledge requires a fine analysis of the tools used, a close investigation of measure and elaboration of data. From Quantum Physics we learned that one should not confuse our instruments, which are specific constructions, nor the result of a measure with an intrinsic organisation of “reality”: when a photon goes through a double slit, we should not “see” a particle if we measure a discrete event out of one slit, or a wave if an interferometer is used. Similarly, projecting “out there” our fantastic arithmetic machines forces a Philosophy of Arithmetic onto an understanding of Nature, with its myth (and strength) of *exactness* and *absoluteness* and its artificial distinction

\* Revised extended abstract of an invited lecture, Proceedings of the *European Software Engineering Conference and ACM SIGSOFT Symposium*, September 1-5, 2003, Helsinki, Finland.

Copyright 2003 ACM 1-58113-743-5/03/000. \$5.00.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

between *software and hardware*. None of these makes sense in today's Natural Sciences.

## 2. CONTINUOUS PHYSICAL PROCESSES

Consider a double pendulum, both the physical device and a computer simulation (easily downloadable from the web). Two parameters give the initial conditions. Push on restart on your preferred computational model, on the same numerical values. And your fantastic arithmetical model will follow *exactly* the same crazy trajectory. Crazy, because the physical device has a chaotic behaviour and so (apparently) has the computer simulation. Yet, arithmetic exactness, our invention, forces the machine to go along the exact same trajectory, by the same round-off, over and over again. Try to iterate a physical double or forced pendulum on the "same" initial values: it will go along a different trajectory. The point is that the physical device is sensitive to initial conditions and that measure, in physics, is always *an interval*: the variation within the measure interval is sufficient to generate paths that differ after a few seconds. There is no such a thing as digital exactness in nature. Yet, this is at the core of computing, our artificial arithmetic invention. You may artificially simulate the measure interval by, say, a pseudo-random generator, but this is just another (one line) program, and the causal structure of the evolution of the physical device and the arithmetic one are essentially different: the trajectories are caused by provably different deterministic structures (see [6]).

Some claim that "all natural processes are computable" (an Extended Church Thesis, so to say). I just gave one, which is not. A physical double pendulum is, mathematically, a *deterministic* machine (its evolution is described, or *determined*, by two equations and, since Poincaré, it is a mathematical issue whether a physical system is deterministic or not). Transform it into an input-output device: a pair of input numbers (degrees, for example) set the initial conditions, up to the physical approximation; the closest integers to the measurable positions of the two heads of the double pendulum after, say, 60 seconds, give the output (today, we can realize pendula with *very little* friction). Save the input-output pairs in a stack. This gives you a function with a finite domain, so a non-intuitionist would still call it computable. Yet, you may extend it to a function defined on all natural numbers. Just work modulo  $180 \times 180$ : when  $(n,m)$  is not already in the stack and  $(n,m) \geq (180,180)$ , start again the machine on  $(n,m)$  modulo  $(180,180)$ . This defines a total number-theoretic function that is provably not computable, by the sensitivity of the double pendulum to initial conditions (exercise: give the details of the definition of the function). Give 3 and 7, say, as input to the physical device; it will start on a (small) interval around these values, not *exactly* on them, and ... who knows where it will end up. No computer simulation would be able to compute (thus predict!) the values that the physical machine will take on these input values. You have very low probability that they will coincide with the computer outputs, as 60 seconds are enough to turn the error *below* your given physical measure or digital approximation, into an enormous, measurable, difference in the trajectory (and final positions). These concepts simply do not exist in an arithmetic machine: the digital data-base is exact. You may need to start one million times your digital model on 3 and 7 before observing a change in the trajectory, due to an hardware bug, typically (the program is too simple to be bugged). The

unpredictability of a sufficiently unstable dynamical system shows up immediately and it is not a matter of complexity, a practical problem of some very large programs, but it is *in principle*, because it depends on the inherent structure of physical measure and on sensitivity to initial conditions.

But, can you sell a double pendulum as a computing device? It is even not a good random generator, as not all its observables have a uniform probability distribution (technically, the right/left positions of the lower head do not need to yield a Bernoulli sequence). Moreover, as a machine, it is not transferable: identical copies will compute different functions. Its software can not be transferred, copied, modified, simply because it does not exist. It is just hardware. This is why we can sell both computer hardware and software, exactly because they are: independent, exact, absolute and programs can be iterated at will and they do what they are expected (predicted!) to do, no more no less (unless they are badly designed.) As Turing soundly observed in his 1950 paper, digital computers are laplacian (exact and predictable) machines; unpredictability, if any, is a matter of practical complexity, it is not a constituent part of their mathematical description, as instability in dynamical systems (see [6] for more on this). So good for exactness.

As for the absolute, some may observe that we are having some problems with concurrent systems: the access to databases may happen to depend on space and time, even on protocols. And we are struggling to avoid this, by implementing semaphores and interleaving techniques or even by advocating homotopy theory (as it is a matter of space-time connectness, see [5]; see also [1]). The other forms of inexactness and lack of absolute are not an issue: hardware or software bugs are there to be eliminated, they are not inherent to the theory, in contrast to approximate and non-absolute measure in modern dynamics and physical theories, with their essential use of *space continua*. Machines invented on the grounds of a philosophy of Arithmetic take up the limits and expressiveness of this exact and absolute theory, with its underlying discrete topology. I am not saying that matter is "continuous", as I avoid essences and ontologies, but just that, since Riemann and Poincaré, we modelled and made the world *intelligible* by smooth manifolds and non-trivial topologies. A laplacian causal regime, in many cases, may at most provide a (very fruitful) imitation (see below).

However, others may claim that integer numbers allow to generate continua, by approximations, Cantor-Dedekind style. Similarly, computer models of physical processes, usually described by analytic and geometric methods, can approximate the analytic solutions in an effective way. This naïve attitude would suggest, say, that any continuously described evolution can be closely followed (shadowed, technically) by a discrete one. This is generally false, except for linear and laplacian systems.

Take your concrete double pendulum and let it go on two physical input values (within two intervals, then). In no way you can prove that your computer simulation, starting on digits within or close to the given interval, will follow its trajectory closely, long enough. Actually, you may prove that, statistically, it will not. The situation is not better if you consider a purely mathematical non-linear system. For example, consider the well-known logistic equation: a version of it is given by the following equation, where  $x'(t)$  is the time derivative and  $2 \leq k \leq 4$ ,

$$x'(t) = k x(t) (1 - x(t)).$$

This is a very simple and well studied continuous model (since Malthus, it models lots of ago-antagonistic processes, including population growth;  $x(0) = x_0$  is taken in  $[0,1]$ ). It may be used to generate a sequence of real numbers (a “trajectory in the continuum”), in  $[0,1]$ , for  $2 \leq k \leq 4$ , under the form

$$x_{n+1} = k x_n (1 - x_n).$$

Fix  $k$  at 4. Can we approximate any sequence of real numbers so generated, by a digital computer? No way, at least not for  $x_0$  in an subset of  $[0,1]$  of measure 1. Even if you fix an  $x_0$  which may be represented exactly in your computer, at the first round-off, the digital sequence and the continuous one will start diverging. When improving your approximation/round-off from  $10^{-14}$  to  $10^{-15}$ , after about 40 iterations the distance between the two sequences will start oscillating between 0 and 1 (the largest possible distance). All you can prove is the following “shadowing lemma” (notice the order of quantification):

*For any  $x_0$  and  $d$  there is an  $\epsilon$  such that, for any  $\epsilon$ -approximated (or rounded-off  $\leq \epsilon$ ) trajectory, there is one in the continuum which goes  $d$ -close to it, at each step.*

Even considering the fortunate case where you can take  $d = \epsilon$  (in some cases this is possible), this boils down to saying that *globally*, your digital sequences are not so wild: they can be actually approximated by a continuous one, or ... there are so many continuous trajectories, that, given a discrete one, you can find a continuous sequence which goes close to it. But, the *reverse* does not hold: i.e., it is not the case that, given an arbitrary continuous trajectory, the computer can approximate it. Various versions of shadowing lemmas hold in sufficiently regular chaotic systems. Yet, several dynamical systems do not yield even weak forms of shadowing results (see [7]).

Let’s summarize the epistemological issue. The point is that we invented the mathematics of the continuum, since Newton and Leibniz, exactly in order to take care of variation and approximation. If Laplace conjecture were correct (in physical processes, a small perturbation induces small consequences), we would have predictability of all deterministic processes as the approximation of the physical measure would be preserved (and the round-off in our digital machines - a small, iterated perturbation - would not affect the simulation). Poincaré showed that Laplace was wrong. Since then, continuous variations and even fluctuations, possibly below the measure interval, came to the limelight as non-negligible *causes* of systems’ evolutions. Frege and Hilbert did not share (to say the least) Poincaré’s natural philosophy and his insight into unpredictability/ undecidability in mechanics and ... kept proposing exact, absolute and complete knowledge, found over Arithmetic (recall Hilbert’s decidability conjectures for formalized mathematics - “non ignorabimus” - ... 20-30 years after Poincaré’s three bodies theorem: an undecidability result for formal systems of equations!). Of course, Riemann’s spaces were the other “delirium”, at least for Frege. Over them, Relativity Theory was found on the relevance of the non-absolute choice of the reference system and measure, far away from the Newtonian absolutes of the philosophy of Arithmetic. In short, both modern Dynamical Systems and Relativity Theory proposed different *causal structures*, w.r.to Laplace’s Physics, by an essential role of the approximate measure and relativized space continua, with no absolutes (variation and

reference systems yield different causal relations; see [2] and [6]). Our laplacian logical machines (in Turing’s words, 1950), a fantastic artefact, derived from and inherited Frege’s and Hilbert’s philosophy and approach, based on Arithmetic. Actually, they were made possible by it, since the perfect iteration of processes is their aim, on exact and absolute date bases: the access to these bases *must not* depend on protocols, on distance, on the path followed ... otherwise we try, by all means, to correct this dependence and enforce exactness and absoluteness - and we can almost reach perfection: observe how effective are the access protocols to the web, independently of paths, distance, speed.... However, in networks and distributed systems, we are facing some difficulties. Besides the enormous logical complexity of today interaction nets, we witness at least a double challenge: the distribution in space, which yields “true” physical time (there is no time without space), with the subsequent issue of synchronisation (as a matter of fact, time is a *relational* matter); the “friction” between systems with different causal structures - one that was invented from Arithmetic computability, the other that we better understand by dynamics in space-time continua (the global environment of the network). And we struggle to reduce the impact of physical space and time onto our modern (distributed) machines in order to have them back to the exact and absolute performances of Turing’s Logic Computing Machine; these were conceived out of physical space and time (TMs have only a linear space - up to obvious equivalences - and their - artificial - time is secreted by the clock). Yet, it is their persisting arithmetic nature, the very reason for their current computational effectiveness, that makes imitation of physical non-laplacian processes a further scientific challenge, as hinted above.

### 3. VARIABILITY AND MATERIAL UNITY IN SOME LIFE PHENOMENA

Structural stability is a core phenomenon in Biology. Yet, variation and variability are at least as important. Forms iterate, but change as well: they are never identical. Evolution is surely based on sudden changes, but continual variation contributes to it as well: never an offspring is *identical* to the parent cell, plant, animal.

As stressed by Edelman in several writings (see [4]), all evolutive systems share another fundamental property, called “degeneracy”. Organs, living units, species ... possess *non-isomorphic* substructures that *may* express themselves in *similar*, but non-identical ways. Different combinations of genes may participate to the development to the same or similar structures; different parts of the brain may contribute to or support the same or a similar function. Of course, lack of identity (isomorphism) in the underlying structure, as well as in the function expressed, makes degeneracy very different from redundancy, a familiar notion in computing. Moreover, the non-identity of structure and expression may lead to variation: a minor chance in the environment may induce a relevant change in the expressed function (form, structure, action), which induces variation or enhances further degenerate substructures, which yield different expressions.

Plasticity is a further form of variability, which adds to degeneracy. Plasticity, which is always present in life, reaches

its highest degree in neural structures: neurons change form, connections, electrostatic fields .... Patterns of connectivity emerge continually: the very hardware components of neural systems are variable and emerging by cell migration, death and differentiation. The point is that in brain there is nothing else but continually and continuously *varying hardware*: the soul/body split and the software/hardware distinction are our own (fantastic) invention to explain thought mystically and have rigid electronic hardware compute.

Let's elaborate some more on this. Regularities that appear along the dynamics of life phenomena are grounded on common or similar hardware, like whirlpools that may have the same shape over different material supports. The formation of analogue (mathematical) patterns on different hardware only witnesses for some common aspects of the bio-chemical or physical structures, and have nothing to do with the independence of software from hardware: these patterns are not *portable*, at most they may similarly *emerge* elsewhere. There is no symbol pushing in the brain, but emerging regularities, in correspondence to action, over similar biological structures. Many try to understand these regularities as attractors, synchronizations of oscillators... as dynamic forms that regulate action. However good this may be, the dynamical approach tries to single out *constitutive laws* for natural phenomena, along an *evolutive path*, in particular for phylogenesis and ontogenesis. At the end of the phylogenetic path, our human communities invented symbolic cultures (languages, alphabets and... portable computer software) in order to enhance communication, also by machines. These cultures are one of the many radical novelties that appeared along evolution, within social groups of animals that were able, first, *to communicate meaning by symbols* and, later, *to detach symbols from meaning* (the phonetic alphabet and its 0-1 encoding) and ... have them run on electronic hardware. These extraordinary inventions of ours should not be projected backwards, as intrinsic forms of the interaction, on amoebas and neurons that were there well before this entire process and that are just (very complex!) material entities: they use no language, as *symbolic activity*, to communicate. The linguistic-computer metaphor, in Neurosciences, may be suggestive, but it is very misleading as the actual challenge is to grasp the functioning of this immensely complex hardware, where nothing else happens but permanent changes of forms and structures, along dynamic connections and exchanges ... including fluid concentrations and proteins' deformations. With no software, this is the key monist assumption. Similarly, the computer analogy in cognition, with its essential software/hardware split, is the latest residual and most extreme form of Cartesian dualism (soul/body). And the portability of software, mentioned above in reference to a dying computer, yields a modern form of metempsychosis (commonly and soundly used in - bad - science-fiction movies).

Of course, in neural systems there may be spikes as well, but they are just carrier's waves. Since spikes, as 0-1 computer-like states, were seen first, many focused on them for (too) long. Yet, the actual process is elsewhere: what appear as states at the electric measure are the carriers of dynamical processes, which include pre- and post-synaptic biochemical cascades, time gaps between potentials, ionic concentrations inside and outside the neurons.... And this dynamicity is grounded on properties that are unique to life: plasticity and degeneracy. Both require the unity of living cells: plasticity, as continuing

change/deformation, is only possible because neurons have a membrane, which give unity and stability while changing; they are subject of "autopoiesis" (they permanently regenerate their own components, while preserving unity and functions). Degeneracy presupposes the integration of non-isomorphic substructures in a whole, by connectivity and upward and downwards chemical cascades (hormones, neurotransmitters ...). Spikes may be even absent. The Paramecium has remarkable cognitive performances and it is ... a one-cell "animal". It coordinates more than 2,000 cilia to swim; a few of them may move to push food towards an opening in the membrane (a "mouth"), in the opposite direction than usual or than the others. No spikes, no apparent action potentials, yet movement and a complex behaviour: apparently, the cascades of deformations of proteins along microtubules are enough to handle the processes required, including, perhaps, some learning abilities (but very little is known). The Caenorhabditis Elegans has about 300 neurons and no action potentials as well: local interactions and induced deformations of neurons and of their biochemical components seem sufficient to its non-obvious cognitive performances (it recently got the Nobel Award ... well, not the CE, but two biologists who worked with it).

Claiming that information and its elaboration is only a matter of 0-1's is an incredible projection onto natural phenomena of our (extraordinary) invention to encode communication between humans (and have rigid arithmetic machines function). But, as animals, we fortunately use also deformations to communicate: face expressions, gestures, caresses. And they are essential to survival, even today. We badly need an intrinsic geometry of information to handle the communication and information processes in Biology, well beyond the dualist, discrete, rigid computational model. Efforts are made within the dynamical approach; morphogenesis is one of its best applications, since Turing's 1952 paper.

Of course, there exist rigid, yet non-dualist, structures. Macromolecules, such as proteins, are usually quasi-fluid, as they permanently change form (their function is in their form and deformations); yet there is a fundamental exception, genes. They are as rigid as stones; this is why inherited information (forms) may last millions years (but degeneracy contributes to make them and their expression subject to variation). Moreover, genes express themselves within proteins' environments and the traces of the past they transmit are embedded in one of the most dynamic processes in life phenomena: ontogenesis. And this interplay is, of course, a key scientific challenge in the formation and development of life: the "material contingency" of the entire process, that is, its dependence on the specific matter that composed it since its origin.

As mentioned several times, some aspects of dynamics in Biology are modelled by Dynamical Systems, one of the main branches of modern Mathematical Physics. Yet, Biological systems presents some further challenges that are badly described by this physical *theory*. They are *mathematically non-stationary*, i.e. the laws governing a process may change along the very process. They witness *phase changes*, as in Physics, but also frequent *changes of phase space*. That is, the very space of pertinent variables and observables changes. Physics provides surely the basis for their analysis, but life phenomena are like a "singularity" of physics and we still do not know how to look inside this singularity. Matter is

unique, of course (I am a monist), but this does not mean that the conceptual and mathematical tools we gave us to understand it are of just one type: even within Physics we are far from unifying the mathematics of Quanta with that of Relativity. And we look for a novel synthesis that would certainly modify both field theories, not “reduce” one to the other. The *dependency* of Biology on Physics is obvious, for a monist, but *reduction*, as commonly understood, is not, as it is a *theoretical operation*: it reduces to an historically given theory. When the gap is huge, like between sub-lunar and supra-lunar phenomena at Newton’s time, science advances by proposing a novel unity (new laws and conceptual structures), new mathematics (infinitesimal calculus), not just reducing to existing theories. Physicists struggle for unification or novel syntheses when working at micro- and astro-physics; even more so it must be done when looking at a stone and an amoeba. Let alone the reduction to arithmetic silicon machines, our latest invention. As computer scientists, we have to stress the originality and depth of our science, its autonomy, in order to contribute to *scientific* interdisciplinarity against the flat, uncritical transfer of our scientific frame onto all sorts of phenomena. And we need a close analysis of what a computational *imitation*, in Turing 1950’s words, actually gives us, with respect to mathematical *modelling* (In short, a *physico-mathematical model* tries to propose, by mathematics, some constitutive properties of the intended phenomenon; a – functional - *imitation* only yields a similar behaviour, typically based on a different causal structure; see [6] for more on this). A closer analysis of this difference may improve both computer imitation, often the best we can do, and mathematical modelling.

The way of looking at the object of study is very important for this aim: from biologists we must learn to focus on variability and action. Brain, say, is not an input-output device, but it constituted itself along evolution for the purposes of *action*: it plastically changes while acting via a body over a changing environment. This is a very different perspective. For example, recent evidence suggests that brain has no static somatotopic encoding of body parts (there is no homunculus representing and shaped like our body, in the brain, as believed for long), but it “mirrors” and anticipates, locally, action, in an intricate fashion: neurons activate not in correspondence of parts of the body (and even less as encoding of the world), but in correspondence of ongoing or expected activities, even of other - similar - bodies (this is particularly evident in the so called “mirror neurons” which activate when an animal moves or sees an akin performing the same movement).

The challenges are many. Let me just recall a few concepts that are still too little analysed, mathematically: plasticity, degeneracy, non-stationarity, autopoiesis ... They are at the core of the singularity and criticality of life, as a physical phenomenon: living beings or species do not follow a geodetics along a given phase space, but they shape

themselves while acting on a changing environment (phase space), which changes also by this very action. There is no such a thing as a biological environment in “equilibrium”, since there is no life without variation. *Pathologies* contribute to the *extended critical states* ([3]) to which life belongs and both notions are unknown to current physical *theories*. We need brand new conceptual structures, mathematics and computational imitations (and models) in order to handle and analyse them rigorously, as when we invented the notion and the mathematics of gravitational field to unify (apparently) different physical phenomena, a falling apple and the motion of the Moon.

## REFERENCES

The many **references** needed for this short paper may be more extensively found in the author's articles, downloadable from: <http://www.di.ens.fr/users/longo>.

- [1] Aceto L., Longo G., Victor B. (eds.) The difference between Sequential and Concurrent Computations. *Special issue*, Mathematical Structures in Computer Science, Cambridge U. Press, vol. 13, n. 4-5, 2003.
- [2] Bailly F., Longo G. “Space, Time and Cognition: From The Standpoint of Mathematics and Natural Sciences” in Causality and Mind (Peruzzi ed.), Kluwer, to appear, 2003 (version française à paraître dans la Revue de Synthèse, Paris, n. 1, 2004).
- [3] Bailly F., Longo G. “Objective and Epistemic Complexity in Biology” Invited lecture, International Conference on Theoretical Neurobiology, *New Delhi*, February 2003a (to appear).
- [4] Edelman G., Tononi G. A Universe of Consciousness. How Matter Becomes Imagination, Basic Books, 2000.
- [5] Goubault E. (ed.) Geometry in Concurrency, *Special issue*, Mathematical Structures in Computer Science, Cambridge U.P., vol.10, n.4, 2000
- [6] Longo, G. “Laplace, Turing and the “imitation game” impossible geometry: randomness, determinism and programs in Turing's test”. Invited lecture, Conference on Cognition, Meaning and Complexity, Roma, June 2002. (version française dans Intellectica, n. 35/2, 2002, avec la réponse aux “peers commentaries”, au sujet des régimes de causalité).
- [7] Sauer T. “Shadowing breakdown and large errors in dynamical simulations of physical systems”, Physical Reviews E 65, 036220, 2002.
- [8] Sini C. Filosofia e scrittura, Roma-Bari, Laterza, 1994.